Writing Cocoa Automator Actions

Session 137

These are confidential sessions—please refrain from filming or taking pictures.

Application Technologies

Writing Cocoa Automator Actions

Kerry Hazelgren
Automator Engineer

http://developer.apple.com/wwdc2005/

Introduction

- What will you learn?
- What is Automator?
- Why should you create Automator actions?

What You Will Learn

- How to design actions
- How to create actions
- How to create conversion actions
- How to debug actions

What Is Automator?

http://developer.apple.com/wwdc2005/

“End-users, I suspect, will quickly discover that Automator is the fulfillment of a dream: at long last, anyone can program the Mac.”

Matt Neuberg, Tidbits

What Is Automator?

- Workflow automation for the masses
- Drag and drop simplicity
- Flexible and extensible
- Large set of shipping actions
- Workflows can be saved as applications or plug-ins

What Is an Action?

- Focused functionality in a small package
- Work with applications or frameworks
- Supported technologies
  - Cocoa
  - AppleScript
  - Shell scripting languages—new in Xcode 2.1
    - e.g. bash, Perl, Python, Ruby

Why Should I Create Automator Actions?

- It can be easy
  - Actions have a small and focused purpose
  - The tools are familiar
- It can be fast
  - Reuse your existing code
- Give users a new way to use your product
  - Put a new UI on your unique functionality
  - Involve your product in user's daily workflow
- You'll be cool
  - Adopt new technologies
  - Generate attention

Designing an Action

- Decide which type of action
  - Are you targeting a scriptable application?
  - Are you targeting a framework or a technology?
  - Are you targeting a command line tool?
- Keep it small
  - No Swiss Army actions
- Act on arrays of data
- Choose meaningful data types
  - Define your own if necessary

Demo

Unsharp Mask workflow

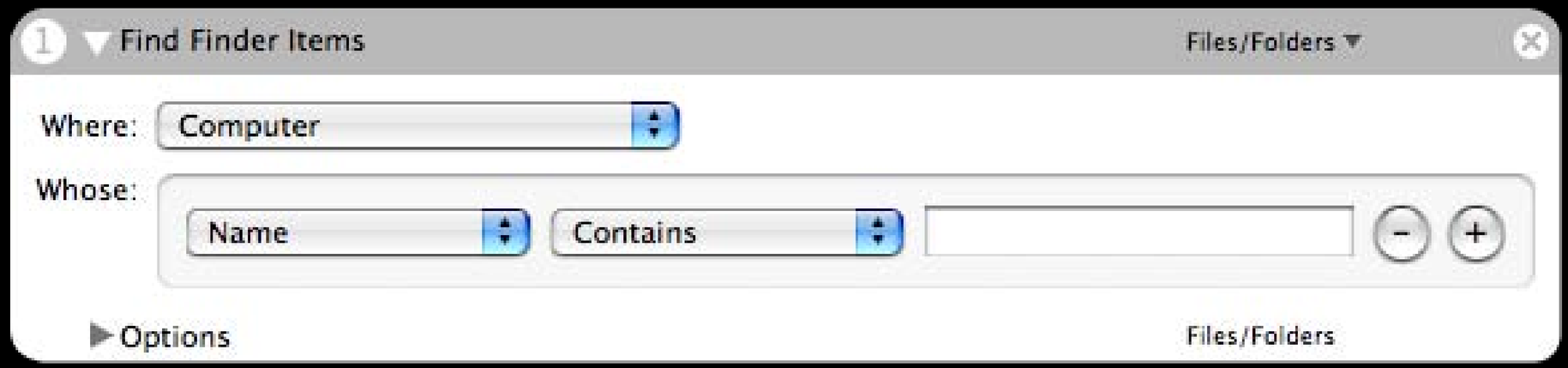
http://developer.apple.com/wwdc2005/

Building an Action's User Interface

http://developer.apple.com/wwdc2005/

Building an Action's User Interface

- Keep it simple
- Follow the published guidelines
  - Use small fonts and controls
  - Minimize vertical space
- Use Cocoa Bindings
- Use shipping actions as examples



Hands-on: Building an Action's UI

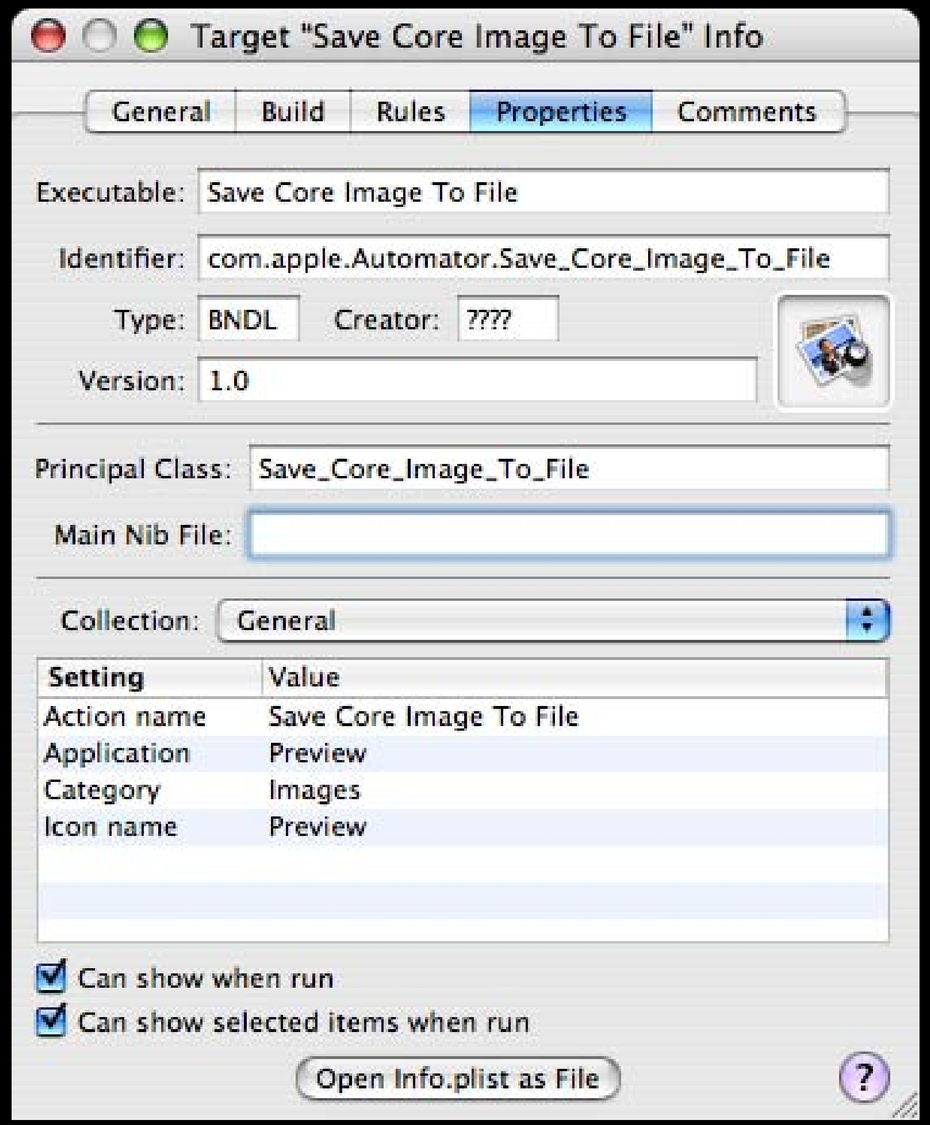
Unsharp Mask (Step 1)

http://developer.apple.com/wwdc2005/

Configuring an Action

Using the Action Target Inspector

- Open the Target Inspector
  - Select “Properties”
- Select desired collection
- Configure properties
- Can also edit the Info.plist file directly



Data Types

- Automator uses UTIs for type identification
  - UTI stands for Uniform Type Identifier
  - Can be used to identify objects, file formats, etc.
  - Used extensively in Tiger
- Automator declares a number of UTIs for you
  - com.apple.cocoa.string
  - com.apple.applescript.object
- Choose an existing UTI or declare your own
  - Use existing types where possible

Defining a New Data Type

- Create a .definition file
  - See Image Definition sample project
- Declare a new UTI type
  - com.yourcompany.yourframework.yourobject
  - com.yourcompany.yourapp.yourobject
- Install in
  - …/Library/Automator
  - Your project's bundle
    - YourApplication.app/Contents/Library/Automator

Hands-on: Configuring an Action

Unsharp Mask (Step 2)

http://developer.apple.com/wwdc2005/

Coding an Action

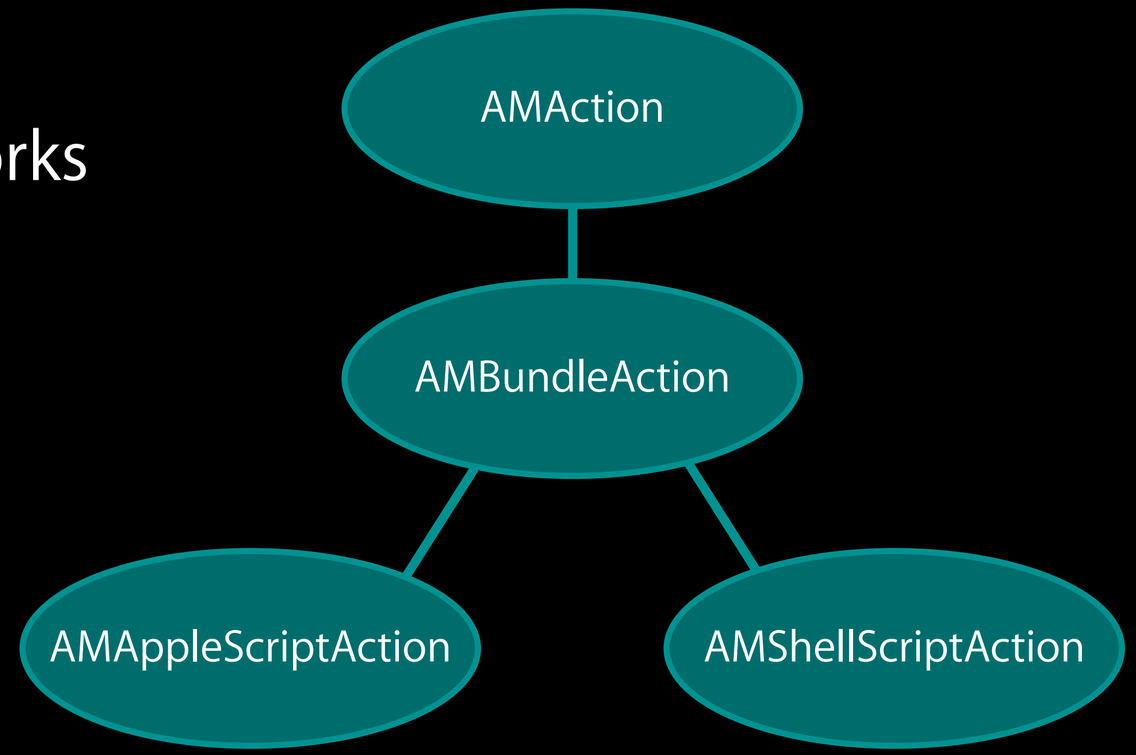
http://developer.apple.com/wwdc2005/

Coding an Action

- Import your existing class files and frameworks
- Override runWithInput:fromAction:error:

Automator Framework

- Defines classes for use in actions
- Located in /System/Library/Frameworks



Tips

- Do not present custom error message dialogs
  - Return your error messages in the error dictionary
- Access your action's bundle with [AMBundleAction bundle]
  - Calling [NSBundle mainBundle] will give you Automator's bundle
- Actions run in a secondary thread
  - Process your user interface on the main thread

```
[self performSelectorOnMainThread:withObject:waitUntilDone:]
```

Commonly Overriden Methods

- Opened
  - Perform the initial setup of your UI
- Activated
  - Keep your UI up-to-date
- ParametersUpdated
  - When your parameters dictionary changes
  - Update your UI if necessary
- UpdateParameters
  - When your UI values change
  - Saves the changes to your parameters dictionary
- Remember to call “super”

Building an Action

- Configure the target
  - Set the product name and extension
- amlint
  - Runs as part of a build phase
  - Looks for common errors
- plutil
  - Can verify the format of your action’s info.plist file

Installing an Action

- Install into any of these directories
  - /Library/Automator
  - ~/Library/Automator
  - /Network/Library/Automator
  - Your project's bundle
    - YourApplication.app/Contents/Library/Automator/
- Not necessary for debugging

Hands-on: Coding an Action Unsharp Mask (Step 3)

http://developer.apple.com/wwdc2005/

Hands-on: Coding an Action

Save Image to File (Step 2)

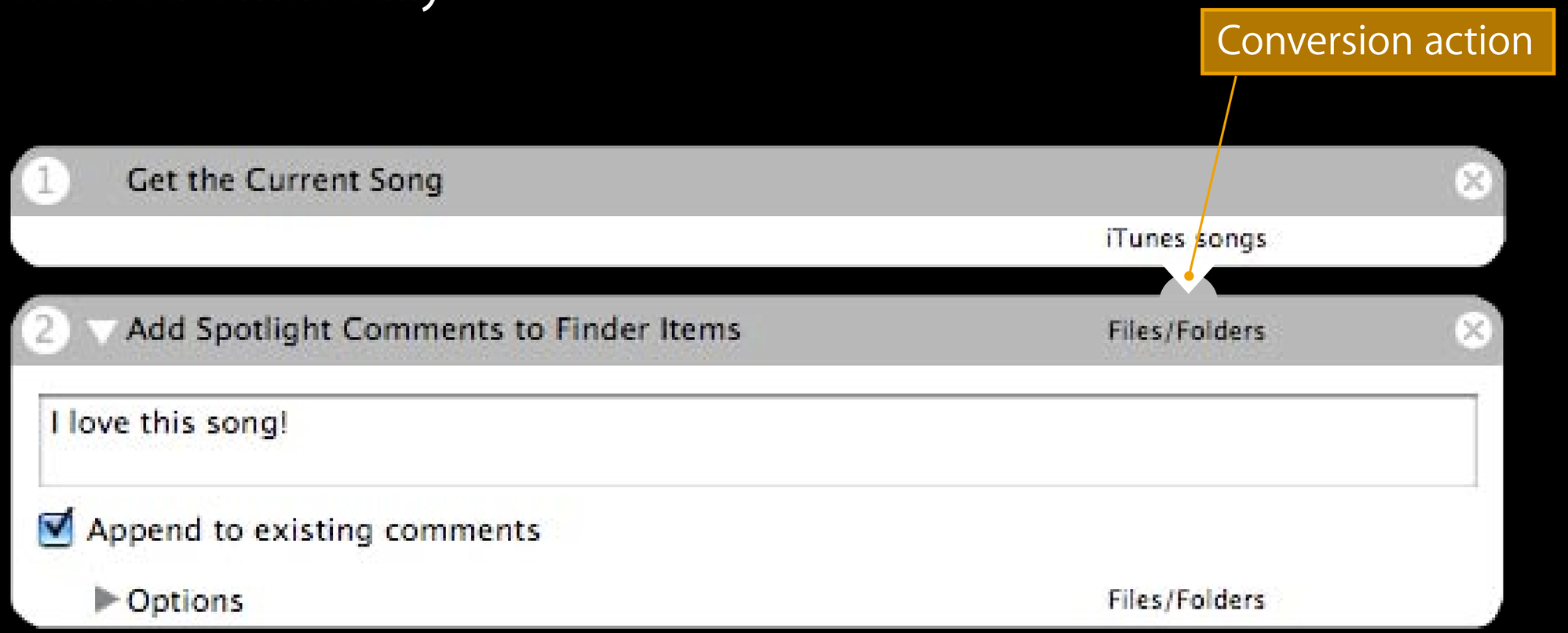
http://developer.apple.com/wwdc2005/

Creating a Conversion Action

http://developer.apple.com/wwdc2005/

What Is a Conversion Action?

- Essential for smooth flow of data
- Can either convert or filter data
- Have no user interface
- Executed automatically



Creating a Conversion Action

- Follow the steps to create an action
- Change the product extension to .caction
- Delete the main.nib

Hands-on: Creating a Conversion Action

Convert Paths to Images (Step 2)

http://developer.apple.com/wwdc2005/

Debugging an Action

http://developer.apple.com/wwdc2005/

Debugging an Action

- Configure the executable
  - Set the action parameter
- Debug as usual
- Use View Results action to monitor your action
- Monitor the workflow log

Hands-on: Debugging an Action Unsharp Mask (Step 3)

http://developer.apple.com/wwdc2005/

What We Learned

- How to design an action
- How to create actions and conversion actions
- How to debug actions
- How easy it really is

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

Mac OS X Lab: Automator Action Development	Session 1021 Friday 10:30AM
Building Automator Actions for System Administrators	Session 645 Friday 5:00PM

Reference Library

Documentation

Automator Programming Guide
ADC Home > Documentation > Apple Applications > Automator

Automator Objective-C Reference
ADC Home > Documentation > Apple Applications > Automator

Uniform Type Identifiers Overview
ADC Home > Documentation > Carbon > Data Management

Cocoa Bindings
ADC Home > Documentation > Cocoa > Design Guidelines

Who to Contact

Sal Soghoian
AppleScript & Automator Product Manager
sal@apple.com

Todd Fernandez
AppleScript & Automator Engineering Manager
trf@apple.com

Automator Developers
Mailing List
automator-dev@lists.apple.com

Q&A

John Montbriand
DTS
montbriand@apple.com

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference